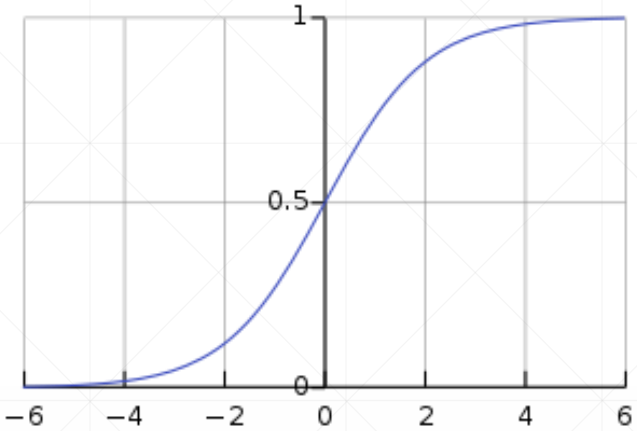
Машины опорных векторов. Обучение без учителя

Гончаров Павел kaliostrogoblin3@gmail.com
Нестереня Игорь nesterione@gmail.com

Повторение

Норма: мера измерения длины вектора. Евклидова норма – $\|w\|^2$, (для простоты условимся, что $L2 = \|w\|$).

Норма для $\|w\|$, где $w \in R^2 = \sqrt{w_1^2 + w_2^2}$



Логистическая функция, сигмоид:

$$\frac{1}{1 + e^{-z}}$$

Гипотеза для логистической регрессии:

$$h_w(x) = \frac{1}{1 + e^{-w^T x}}$$

Целевая функция для логистической регрессии:

$$-\frac{1}{m}\sum_{i=1}^{m} ylog\big(h_w(x)\big) + (1-y)\log(1 - h_w(x))$$

Машина опорных векторов, SVM

Задача логистической регрессии – **минимизировать значение целевой функции J(w)**. Для случая с регуляризацией:

$$\min_w \left(-\frac{1}{m}\sum_{i=1}^{m}\underbrace{\left[ylog\left(h_w(x)\right) + (1-y)\log\left(1-h_w(x)\right)\right]}_{A} + \mathrm{C}\underbrace{\sum_{j=1}^{n} w_j^2}_{B} \right) \rightarrow A + CB$$

Для машины опорных векторов формула примет вид:

$$\min_w \left(C\sum_{i=1}^{m}\left[y\boldsymbol{cost_1}(x) + (1-y)\boldsymbol{cost_0}(x)\right] + \sum_{j=1}^{n} w_j^2 \right) \rightarrow CA + B$$

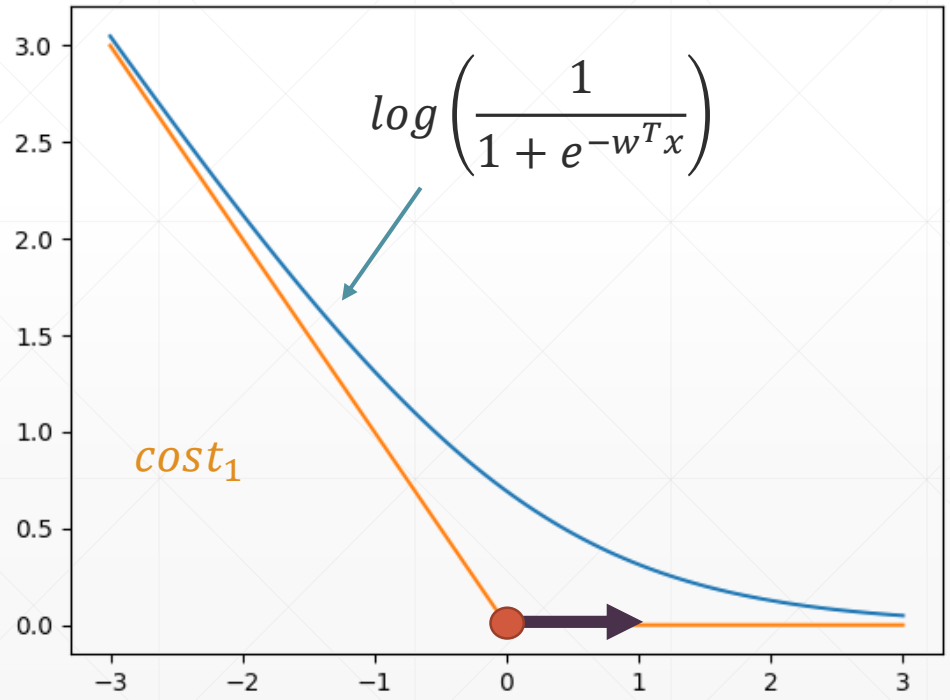
Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве.

Гипотеза и функция ошибки в SVM

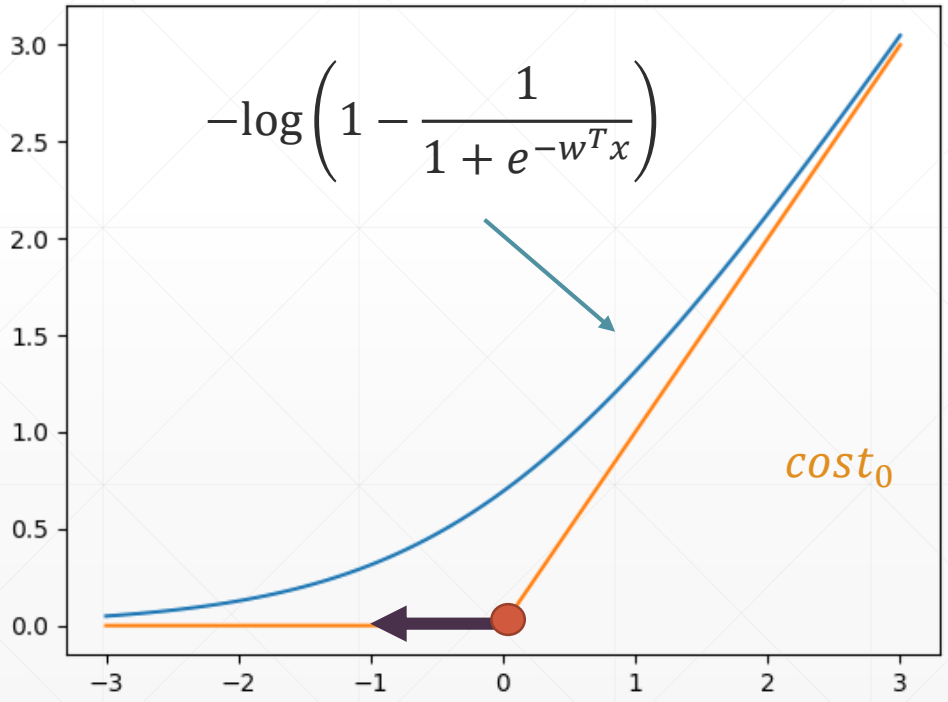
Гипотеза в SVM – это простой пороговый классификатор: 1, если $w^T x \geq 0$ и ноль в противном случае.

$$h_w(x) = \begin{cases} 1 & \text{если } w^T x \geq 1 \\ 0 & \text{если } w^T x \leq -1 \end{cases}$$

$$cost_1(x) = \max(0,1 - w^T x)$$



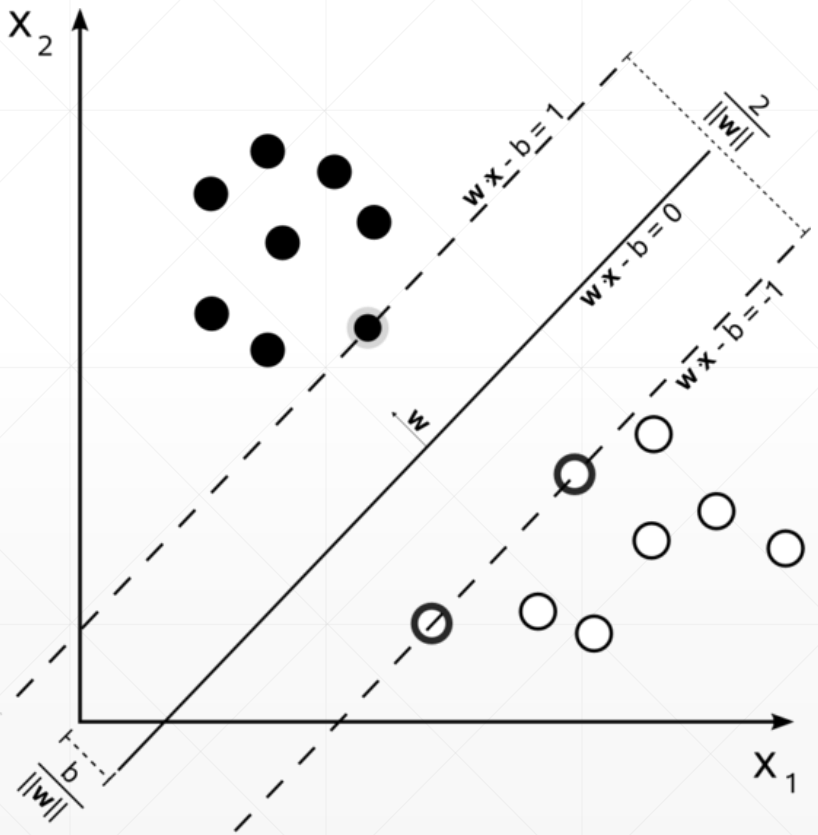
$$cost_0(x) = \max(0,1 + w^T x)$$



- **Получается, что если у должен быть равен 1, то мы хотим, чтобы $w^T x \geq 1$**
- **Если же у должен равняться нулю, то мы хотим, чтобы $w^T x \leq -1$**

Метод классификатора с максимальным зазором

$$J(w) = \min_w \left(C \sum_{i=1}^{m} [y\boldsymbol{cost_1}(x) + (1-y)\boldsymbol{cost_0}(x)] + \sum_{j=1}^{n} w_j^2 \right) \rightarrow CA + B$$



В ходе минимизации мы пытаемся свести эту часть к 0

Если первая часть выражения, A, будет равняться нулю, то мы приходим к задаче минимизации нормы вектора параметров:

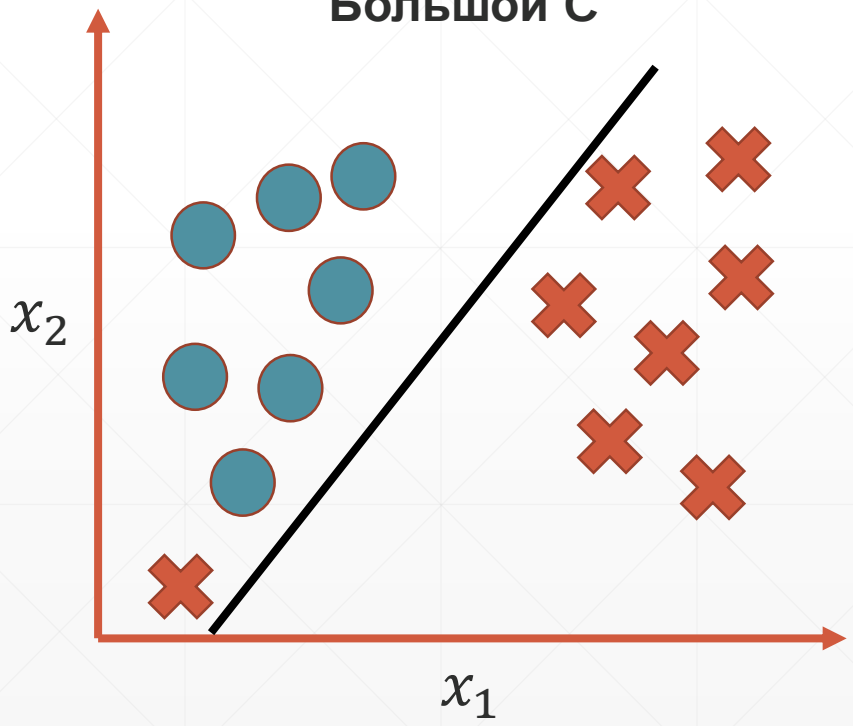
$$J(w) = \min_w \sum_{j=1}^{n} w_j^2 \rightarrow \min_w \|w\|$$

$$\text{при } h_w(x) = \begin{cases} 1 & \text{если } w^T x \geq 1 \\ 0 & \text{если } w^T x \leq -1 \end{cases}$$

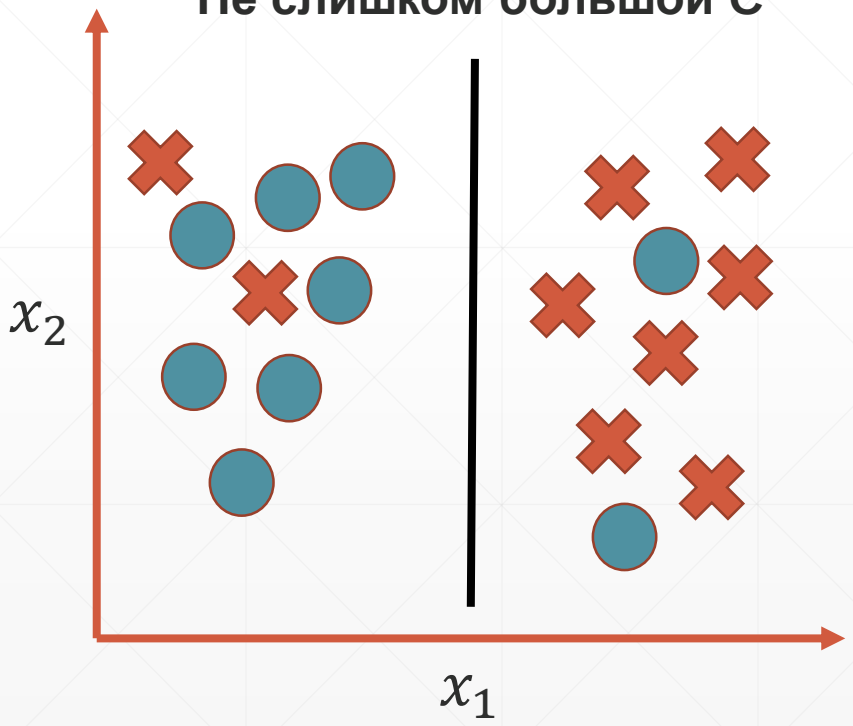
На что влияет параметр регуляризации C

$$J(w) = \min_{w}\left(\boxed{C}\sum_{i=1}^{m}[y\boldsymbol{cost_1}(x) + (1-y)\boldsymbol{cost_0}(x)] + \sum_{j=1}^{n} w_j^2\right) \rightarrow \boxed{C}A + B$$

Большой C



Не слишком большой C



Модель, у которой не слишком большой параметр C устойчива к выбросам в данных

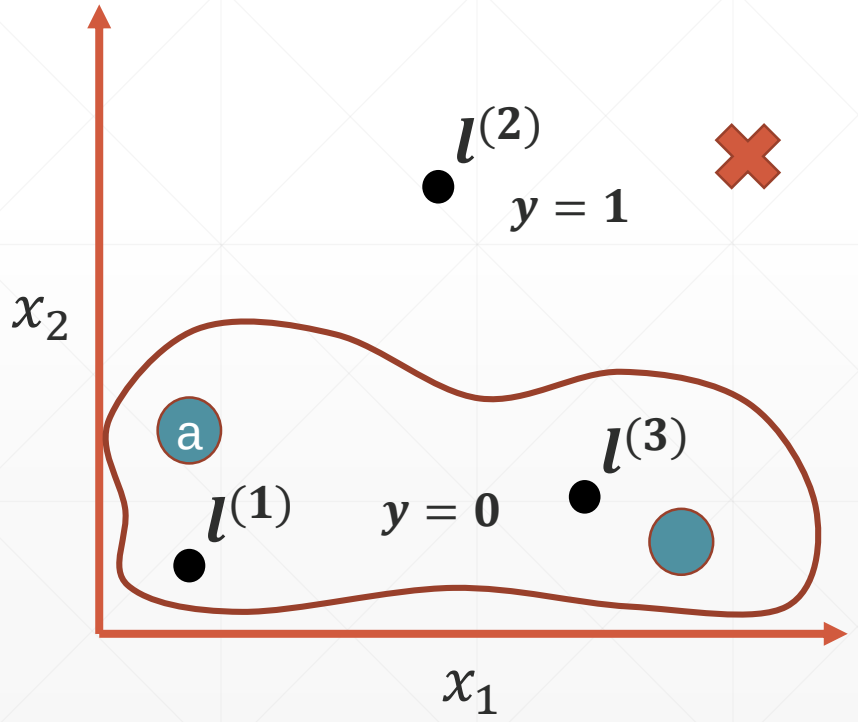
Ядро машины опорных векторов, kernel

Для того, чтобы описать данные сложной формы с нелинейной зависимостью, нужно использовать полином – $w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 \dots w_nx_1^n$. Однако подбор нужного полинома и вычисление высоких степеней – **слишком дорого.**
Идея машин опорных векторов заключается в том, чтобы заменить все признаки *X* на признаки *f:*

$$f = similarity\left(x, l^{(i)}\right), \qquad f \in R^n$$

Similarity – это расстояние от X до точки $l^{(i)}$, ядро.

Пример. $l \in R^3$ – это вектор из трех значений. У нас есть всего два класса: кружочки (отрицательный класс) и крестик (положительный класс). $f = 1$, если *x* рядом с $l^{(i)}$.

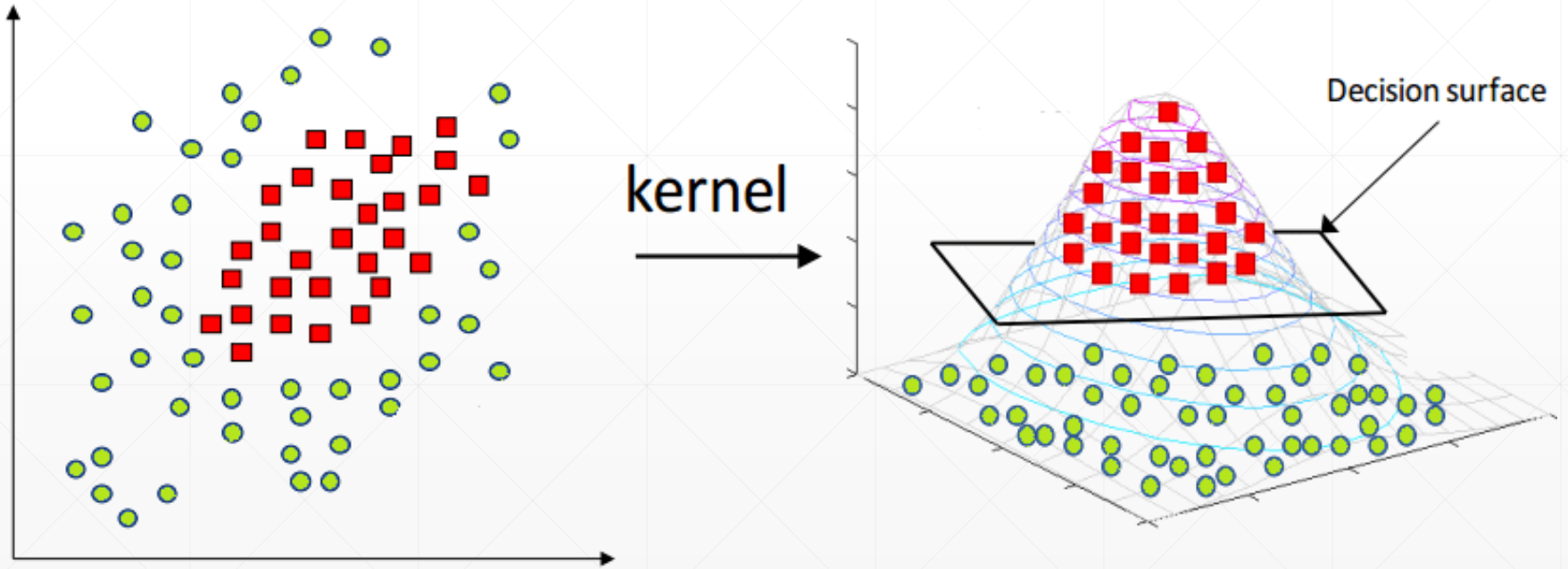


Есть гипотеза: $h_w(x) = w_0 + w_1f_1 + w_2f_2 + w_3f_3$,
где $\boldsymbol{w_0 = 0.5, w_1 = -1, w_2 = 0, w_3 = -1.}$

Для точки ***a*** расстояние до $l^{(1)}$ будет минимальным, значит $f_1 = 1$, а $f_2 = f_3 = 0$.
Тогда $h_w(x) = 0.5 - 1 < 0 \rightarrow y = 0$, т.е. мы *a* относится к отрицательному классу

Ядро машины опорных векторов, kernel

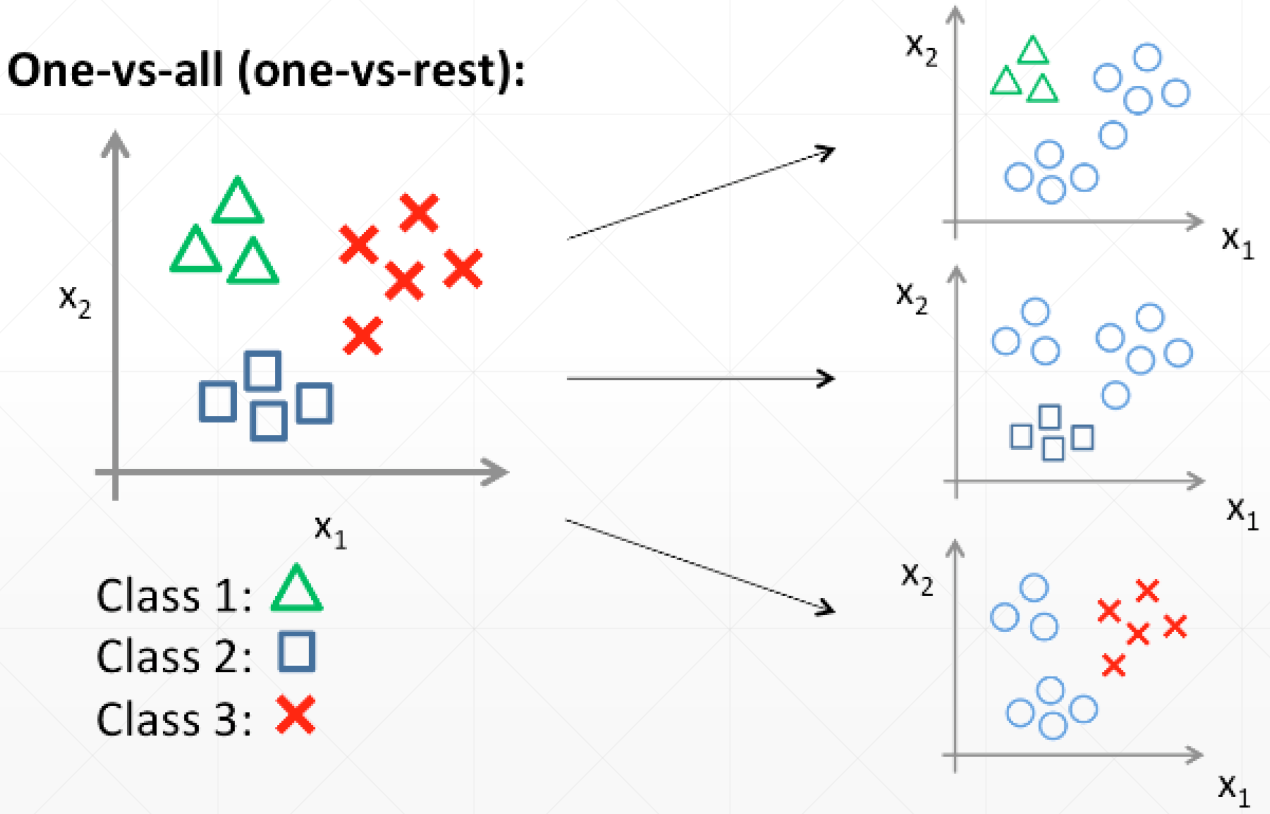
Ядра бывают разных типов: радиально-базисные, линейные, полиномиальные, гауссианами и др.
Выбирать число точек l для измерения расстояний **не нужно**. Как правило, точек l столько же, сколько и примеров в данных, т.е. признаки f – это **расстояние от каждой точки до каждой другой**.



Сверху пример для ядра в виде гауссиана. С помощью новых признаков f мы приводим данные к такому виду, чтобы их было легко разделить.

SVM для многоклассовой классификации

Для того, чтобы применить SVM в случае множества классов (>2), нужно воспользоваться стратегией «один против всех» (one vs all).



Идея в том, чтобы обучить для каждого класса свой классификатор, который будет разделять этот класс от других. После обучения всех классификаторов, нужно объединить их границы принятия решений (decision boundaries).

Классификация алгоритмов по наличию лейблов

Все алгоритмы машинного обучения можно разделить на три группы, в зависимости от того, присутствуют ли лейблы, есть ли верные ответы, в выборке или нет. Эти группы:

- обучение с учителем (supervised learning)
- обучение без учителя (unsupervised learning)
- смешанное обучение (semi-supervised learning)

Все методы и алгоритмы, которые мы рассматривали ранее – это обучение **с учителем**.

К задачам обучения без учителя относятся **кластеризация, сжатие данных, поиск аномалий, визуализация и др.**

Смешанное обучение или полу-контролируемое обучение нужно для тех случаев, когда у нас какая-то часть выборки размечена и на основе этих знаний, мы пытаемся разметить оставшуюся часть данных.

Задача кластеризации

Кластеризация – это разделение множества входных векторов на группы (кластеры) по степени «схожести» друг на друга.

Основная проблема кластеризации заключается в том, что **заранее неизвестно, сколько всего может быть подмножеств данных.**

Для того, чтобы сравнивать два объекта, необходимо иметь **критерий**, на основании которого будет происходить сравнение. Как правило, таким критерием является **расстояние между объектами (среднеквадратичное расстояние, норма, гауссиан и др.).**

- Для вычисления признаков SVM используется ядро, позволяющее оценить расстояние между примерами в обучающей выборке.
- Модель регрессии проверяют, измеряя среднеквадратичное (евклидово) расстояние. Это позволяет оценить, насколько построенная гипотеза верно описывает данные.

Метод K-средних

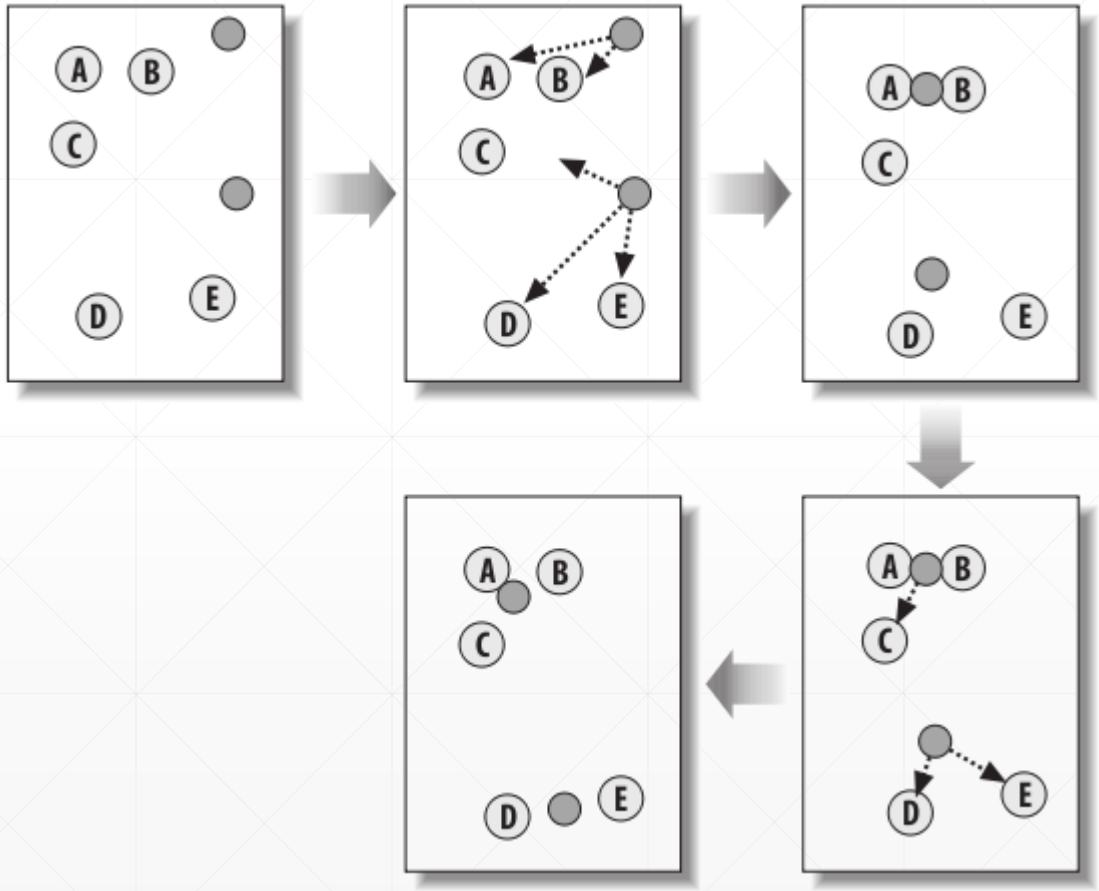
Существует множество методов кластеризации, один из которых – метод K-средних.

Алгоритм.

1. Определяется число кластеров, *m*
2. Выбирают первые центры кластеров
3. Для каждой точки найти расстояние до ближайшего центроида, L2
4. Отнесите каждую из точек к ближайшему к ней центру кластера
5. Вычислите новые центры кластеров:

$$c_i = \left(\frac{1}{n_i}\right)\sum_{j=1}^{n_i} x_i$$

6. Пункт 3
7. Повторять, пока центры кластеров не перестанут изменяться



Алгоритм очень чувствителен к выбору начальных центров кластеров!

Пример работы алгоритма: http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Спасибо за терпение!